

Article

FTIR-SpectralGAN: A Spectral Data Augmentation Generative Adversarial Network for Aero-Engine Hot Jet FTIR Spectral Classification

Shuhan Du ¹, Yurong Liao ¹, Rui Feng ¹, Fengkun Luo ²  and Zhaoming Li ^{1,*} 
¹ Department of Electronic and Optical Engineering, Space Engineering University, Beijing 101416, China; shuhan_du@hgd.edu.cn (S.D.); liaoyr@hgd.edu.cn (Y.L.); fengrui@hgd.edu.cn (R.F.)

² Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China; luofk@aircas.ac.cn

* Correspondence: lizm@hgd.edu.cn

Abstract: Aiming at the overfitting problem caused by the limited sample size in the spectral classification of aero-engine hot jets, this paper proposed a synthetic spectral enhancement classification network FTIR-SpectralGAN for the FT-IR of aeroengine hot jets. Firstly, passive telemetry FTIR spectrometers were used to measure the hot jet spectrum data of six types of aero-engines, and a spectral classification dataset was created. Then, a spectral classification network FTIR-SpectralGAN was designed, which consists of a generator and a discriminator. The generator architecture comprises six Conv1DTranspose layers, with five of these layers integrated with BN and LeakyReLU layers to introduce noise injection. This design enhances the generation capability for complex patterns and facilitates the transformation from noise to high-dimensional data. The discriminator employs a multi-task dual-output structure, consisting of three Conv1D layers combined with LeakyReLU and Dropout techniques. This configuration progressively reduces feature dimensions and mitigates overfitting. During training, the generator learns the underlying distribution of spectral data, while the discriminator distinguishes between real and synthetic data and performs spectral classification. The dataset was randomly partitioned into training, validation, and test sets in an 8:1:1 ratio. For training strategy, an unbalanced alternating training approach was adopted, where the generator is trained first, followed by the discriminator and then the generator again. Additionally, weighted mixed loss and label smoothing strategies were introduced to enhance network training performance. Experimental results demonstrate that the spectral classification accuracy reaches up to 99%, effectively addressing the overfitting issue commonly encountered in CNN-based classification tasks with limited samples. Comparative experiments show that FTIR-SpectralGAN outperforms classical data augmentation methods and CVAE-based synthetic data enhancement approaches. It also achieves higher robustness and classification accuracy compared to other spectral classification methods.

Keywords: infrared spectrum measurement; FTIR; aero-engine hot jet; GAN; data augmentation



Academic Editors: Xia Wang and Kun Gao

Received: 6 January 2025

Revised: 10 March 2025

Accepted: 12 March 2025

Published: 16 March 2025

Citation: Du, S.; Liao, Y.; Feng, R.; Luo, F.; Li, Z. FTIR-SpectralGAN: A Spectral Data Augmentation Generative Adversarial Network for Aero-Engine Hot Jet FTIR Spectral Classification. *Remote Sens.* **2025**, *17*, 1042. <https://doi.org/10.3390/rs17061042>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Infrared spectral classification of the hot jet emitted by aero-engines is a critical component in the field of infrared target detection and identification. After combustion in an aero-engine, different gas components and emissions form hot jets. The molecules of the substances in the hot jets vibrate and rotate to form specific infrared spectra. Spectral

information contains data about the combustion process and fuel composition of the aero-engine. By analyzing and classifying the infrared spectra of hot jets from different types of aero-engines and using infrared spectroscopy technology to analyze the specific spectral patterns formed by the molecules of substances under the action of infrared radiation, it is possible to analyze and determine the chemical bonds and structures of the substances and thereby achieve the classification of aero-engines.

Due to the scarcity of public datasets, in the present paper, the FTIR spectral data of the hot jets of six different types of aero-engines were measured through field experiments. In previous studies, there was a contradiction between the demand of deep learning methods for training a large number of learnable parameters and the limited samples in existing datasets. Once existing models reached a certain classification accuracy, it was difficult for them to learn new features to further improve classification accuracy. Therefore, enhancement of the performance of the model under the condition of limited samples became the focus of this paper's research. Similar to the challenges in hyperspectral image classification (HSIC), the measurement cost of labeled datasets was expensive. The solutions (Li [1]) included data augmentation, transfer learning, and unsupervised/semi-supervised feature learning. The limitation of the dataset size affects the generalization ability of the spectral classification model. Data augmentation and synthetic data augmentation methods could improve the model's adaptability to data under different conditions, enhance the generalization ability of the model, improve the robustness of the model, and increase training efficiency. Therefore, data augmentation and synthetic data augmentation methods were investigated in the current paper. Shi [2] proposed a spectral encoding enhancement representation (SEER) method based on phase vectors for enhancing specific and subtle spectral differences when enhancing hyperspectral fluorescence images. Nalepa [3] enhanced the data during the inference stage and constructed an enhancement method based on principal component analysis. GANs were an effective method for data augmentation. Wang [4] reviewed the current development trends of GANs. The basic GAN models included GAN [5], CGAN [6], DCGAN [7], SNGAN [8], and styleGAN [9]. In HSIC, He [10] used a 3D bilateral filter (3DBF) to extract spectral-spatial features and improved the classification performance by combining GANs with semi-supervised learning. Zhan [11] designed a semi-supervised framework (HSGAN) for HSI data using one-dimensional GAN. Zhu [12] used 1D-GAN and 3D-GAN for hyperspectral classification. Ding [13] integrated 2DCNN and GAN for feature extraction, followed by Support Vector Machine (SVM) for classification. Ranjan [14] utilized Wasserstein GAN with gradient penalty to generate high-quality synthetic hyperspectral data cubes, which were then classified using a Convolutional Long Short-Term Memory (ConvLSTM) classifier to enhance classification performance. Wang [15] developed an improved Adaptive DropBlock-enhanced Generative Adversarial Network (ADGAN) to address the challenge of limited training samples. Besides GAN methods, Li [16] used common data augmentations (flipping, rotation, and noise) to reach the upper limit of the enhancement effect when the sample size was tripled and designed a pixel-block pair CNN (PBP-CNN) to improve the performance of the classification network by extracting PBP features and combining decision fusion for data augmentation. Haut [17] adopted random occlusion for data augmentation of CNN in HSI. Wang [18] conducted spectral enhancement in the spectral domain of HSI using a similar spectral construction method. Gao [19] reduced the difference between the augmented samples and the original samples by training the Siamese network through dynamic data selection. In previous studies [20], a CNN with three Cov1D layers was proven to be effective on spectral data and could achieve a spectral classification accuracy of 96%, but the CNN had overfitting problems on the current spectral dataset. To solve the problem of the classification accuracy

of the network model reaching a bottleneck under limited samples, in the present paper, a spectral classification network FTIR-SpectralGAN was constructed based on DCGAN.

FTIR-SpectralGAN primarily processes 1D infrared spectra. This network architecture comprises a generator and a discriminator. The generator includes six Conv1DTranspose layers, with five of these layers integrated with Batch Normalization (BN) and LeakyReLU activation functions. The discriminator employs a multitask dual-output structure, consisting of three Conv1D layers combined with LeakyReLU activations and Dropout regularization. During training, strategies such as unbalanced training, weighted mixed loss, and label smoothing regularization are implemented to enhance the network's performance. Specifically, each discriminator training iteration uses a 1:1 ratio of newly synthesized data to original data. Experimental results demonstrate an accuracy rate of 99%, effectively mitigating overfitting issues commonly encountered by CNNs when dealing with limited spectral samples.

The contribution summary of this paper encompasses the following key aspects:

1. The study employs outfield experiments to conduct infrared spectroscopy measurements on the hot jets of six types of aero-engines. Given that materials possess selective absorption capabilities for infrared radiation, utilizing the infrared spectra of hot jets as data support for the classification of aero-engines is scientifically sound and reliable.
2. This research utilizes an improved FTIR-SpectralGAN network based on DCGAN to address overfitting issues caused by limited sample sizes in infrared spectrum classification. Specifically, FTIR-SpectralGAN adopts 1D processing tailored to the data format of infrared spectra, diverging from traditional 2D operations. In terms of training strategy, an unbalanced approach is implemented where the generator undergoes initial training. Following each update of the discriminator, the generator parameters are optimized five times, effectively mitigating mode oscillation during adversarial training. Additionally, a weighted mixed loss strategy is employed with greater emphasis placed on classification loss to enhance the discriminator's classification capability. Label smoothing regularization is also adopted, setting the label of real samples to 0.9 and combining it with a soft label strategy for generated samples set at 0.1.
3. The paper conducts experiments using both classic data augmentation methods (such as rotation, scaling, translation, resampling, mirroring, jittering, and discarding) and the deep learning-based data augmentation method CVAE, comparing their performance on spectral datasets. In addition, classical spectral feature extraction methods, including one-dimensional convolutional neural networks (1DCNNs), principal component analysis (PCA), and CO₂ feature vectors, were incorporated into the comparison experiment alongside the classifier.

Compared to the classical GAN and DCGAN models, the generator of FTIR-SpectralGAN employs a deeper convolutional architecture and incorporates Gaussian noise. The discriminator utilizes a dual-task framework for authenticity discrimination and six-class classification. In terms of the loss function, a combination of traditional cross-entropy loss with label smoothing is adopted, employing a weighted hybrid loss strategy. Unlike CGAN, the classification task is seamlessly integrated into the discriminator, eliminating the need for additional conditional inputs to the generator.

The structure of the thesis is composed of five parts. The first section reviews data augmentation and GAN methods, briefly presents the methodology, contributions, and framework of this article. The second section introduces the classification principle of aero-engine hot jets, the field experiment design for aero-engine spectral measurement, the design of the dataset, and elaborates on the detailed structure and algorithmic details of the

FTIR-SpectralGAN designed in the current paper. The third section conducts experiments and analysis of the results and uses evaluation indicators to evaluate and analyze the experimental results. The fourth section gives the discussion on the results. The fifth section summarizes this paper.

2. Material and Methods

Section 2 begins with an introduction to the principle of spectral classification of aero-engine hot jet, then proceeds to describe the composition of aero-engines, the composition of hot jet, and the principle of infrared spectrum generation. It briefly outlines the design of the spectral measurement experiment and elaborates on the production and preprocessing of the spectral dataset. Then, the spectral classification network structure design method is given.

2.1. The Principle of Aero-Engine Hot Jet Spectral Classification

The experimental objective of this paper is the aero-engine commonly used in modern aircraft, including two types: turbojet engines and turbofan engines. Their basic structures are all composed of an air inlet, a compressor, a combustion chamber, a turbine, and an exhaust nozzle. The schematic diagram of aero-engine structure is shown in Figure 1.

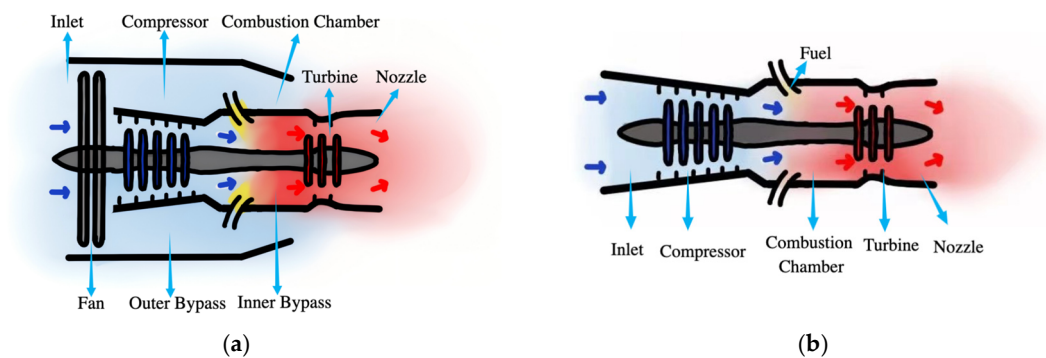


Figure 1. Schematic diagram of the structure of jet aero-engines: (a) represents the turbofan engine; (b) represents the turbojet engine.

The composition of the hot jet from an aero-engine is determined by the combustion process and mainly includes chemical substances such as carbon dioxide (CO_2), steam (H_2O), carbon monoxide (CO), nitrogen oxides (NO_x), unburned hydrocarbons (HC), and particulate matter (PM). Analyzing the composition of the hot jet can help identify the classification characteristics of aero-engines.

Infrared spectroscopy is a commonly used method for quantitative analysis of substances and identification of compound structures. When a substance is exposed to infrared radiation, molecules selectively absorb radiation of specific frequencies, causing molecular vibrations or rotations, resulting in changes in dipole moments and transitions between energy levels. The intensity of the transmitted light in these absorption regions weakens and can be recorded as an infrared spectrum of wavenumber versus transmittance [21–23].

The infrared radiation emitted by the hot substances in the aero-engine hot jet can be measured non-contact through an FTIR spectrometer, obtaining a spectrum with different frequency peaks. The infrared spectrum of the hot jet represents the transitions of chemical substances between different energy levels. Gases are composed of atoms and molecules, and the energy of both is the sum of translational energy and internal energy, with the internal energy consisting of a series of specific and discrete energy levels. A schematic diagram of the quantized energy levels of a molecule is shown in Figure 2.

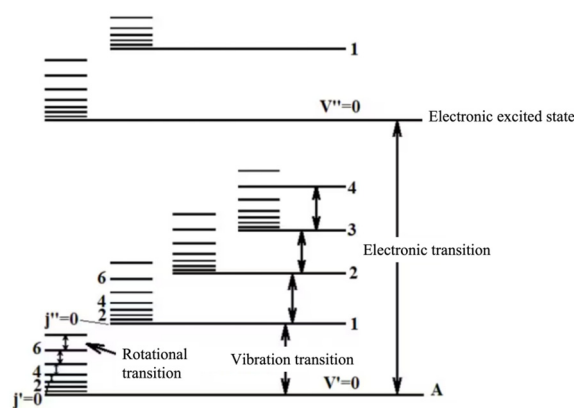


Figure 2. Schematic diagram of the quantized energy levels of a molecule (V represents the energy level and J represents the energy).

The selective absorption of infrared radiation resulting from molecular vibration and rotation is manifested as specific absorption peaks of substances for infrared radiation of different wavelengths. The selection rule that must be satisfied for a molecular transition upon absorption of infrared radiation can be obtained by solving the Schrödinger equation for the energy value E_V of the V energy level:

$$E_V = \left(v_i + \frac{1}{2} \right) h f, v_i = 0, 1, 2, \dots \quad (1)$$

where h is the Planck constant, f is the vibration frequency, and v_i is the vibration quantum number of the i th mode. The energy difference between two adjacent energy levels is $\Delta E_V = h f$.

The selective absorption of infrared radiation by different molecules can be used to determine the molecular structure by analyzing their characteristic absorption peaks. Similarly, for the infrared spectra of hot jets from different types of aero-engines, there are corresponding spectral features that support the classification of aero-engines. Therefore, this paper conducts data analysis on gas infrared spectra, focusing on the mid-infrared region of fundamental vibration frequencies, to find the classification features of the infrared spectra of aero-engine hot jets.

2.2. Spectral Dataset

2.2.1. Experimental Design for Aero-Engine Spectral Measurement

The paper employs the approach of field measurement to collect the infrared spectral data of the hot jets of different types of aero-engines. The specific parameters of the two telemetry FTIR spectrometer devices used in the experiment are provided in Table 1.

Table 1. Parameters for FTIR spectrometers.

Name	Measurement Pattern	Spectral Resolution (cm^{-1})	Spectral Measurement Range (μm)	Full Field of View Angle
EM27	Active/Passive	Active: 0.5/1; Passive: 0.5/1/4	2.5~12	30 mrad (no telescope) (1.7°)
Telemetry Fourier Transform Infrared Spectrometer	Passive	1	2.5~12	1.5°

The experiment simultaneously measured the hot jet of the aero-engine through two telemetry FTIR spectrometers. The layout of the field experiment is shown in Figure 3.

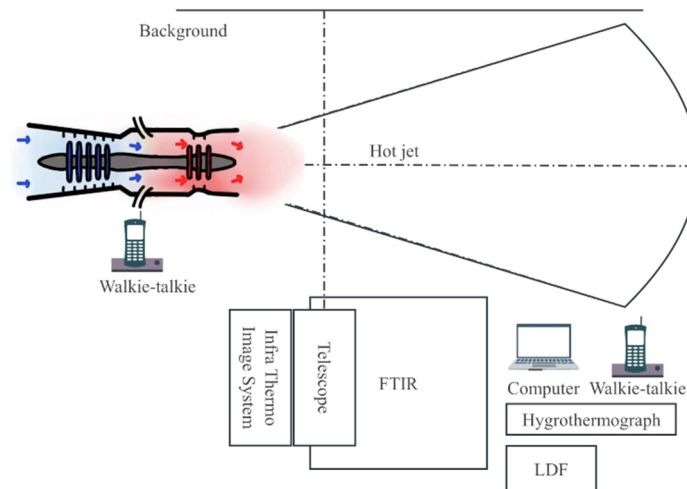


Figure 3. Layout diagram of the spectral experiment of the aeroengine hot jet.

The two spectrometers in the experiment were positioned at a distance of five to ten meters from the aero-engine, being capable of conducting measurements in the hot jet region behind the tail nozzle. This ensured that the signal energy could fill the field of view of the spectrometer and acquire measurement results with a high signal-to-noise ratio. Under these ground test conditions, the experimental distance was relatively short, and the temperature of the hot jet gas differed significantly from the background. It was temporarily assumed that there was no influence from the atmospheric background.

2.2.2. Spectral Dataset Production

The information of the aero-engine hot jet spectral dataset collected in the field test is shown in Table 2 as follows.

Table 2. The six types of aero-engine hot jet spectral dataset.

Serial Number	Class	Type	Number of Data Pieces	Full Band Data Volume
1	C 0	Aero-Engine 1 (Turbofan)	256	16384
2	C 1	Aero-Engine 2 (Turbojet)	48	16384
3	C 2	Aero-Engine 3 (Turbofan)	712	16384
4	C 3	Aero-Engine 4 (Turbojet)	199	16384
5	C 4	Aero-Engine 5 (Turbojet)	380	16384
6	C 5	Aero-Engine 6 (Turbojet)	193	16384

During the experiment, real-time communication was maintained with the responsible personnel of the aeroengine via the intercom system. It was required that any changes in speed be promptly reported in real time and each test speed be stabilized for at least 1 min whenever possible. Due to varying conditions during the aeroengine tests, the quantity of data acquired by each model differed. The environmental temperature and humidity were also recorded during the experiment, as detailed in Table 3.

Table 3. Environmental factors of the experiment.

Serial Number	Class	Environmental Temperature	Environmental Humidity	Detection Distance
1	C 0	19 °C	58.5% Rh	5 m
2	C 1	16 °C	67% Rh	5 m
3	C 2	14 °C	40% Rh	5 m
4	C 3	30 °C	43.5% Rh	11.8 m
5	C 4	20 °C	71.5% Rh	5 m
6	C 5	19 °C	73.5% Rh	10 m

The experimental instrument utilized was an FT-IR spectrometer equipped with blackbody radiation calibration. The DN value of the interference digital signal, which reaches the detector, was collected through laser signal zero-crossing sampling. For a continuous light source, the monochromatic interferogram equation was integrated to determine the signal intensity $I(\delta)$ detected by the detector at an optical path difference of δ .

$$I(\delta) = \int_{-\infty}^{+\infty} B(\nu) \cos(2\pi\nu\delta) d\nu \quad (2)$$

In this context, ν represents the wavenumber, δ denotes the optical path difference, and $B(\nu)$ signifies the wavenumber-dependent optical intensity function after instrument correction. The instrument correction accounts for factors such as the efficiency of the beam splitter, detector response, and the amplification effect, which can be regarded as a constant factor, also known as the correction factor. $I(\delta)$ is the integral over all wavenumbers ν , representing the cumulative light intensity across different wavelengths. By continuously varying δ , a complete interferogram can be obtained through integration.

The inverse Fourier transform of the interference signal offers the light intensity $B(\nu)$ in wave number (cm^{-1}):

$$B(\nu) = \int_{-\infty}^{+\infty} I(\delta) \cos(2\pi\nu\delta) d\delta \quad (3)$$

The spectral interferogram $I(\delta)$ is called the cosine Fourier transform of the spectrum $B(\nu)$, and $B(\nu)$ is also the inverse Fourier transform of the cosine of $I(\delta)$.

When conducting research on gas detection and identification using passive FTIR spectroscopy, since the emissivity of most substances in nature as the background is very high, the method of calculating the brightness temperature ([K]) [24,25] spectrum with a constant baseline can be employed for detection. The use of the brightness temperature spectrum eliminates the need for pre-measurement of the background spectrum and enables the direct extraction of target gas features from the brightness temperature spectrum. The brightness temperature of an actual object is the temperature of a blackbody at the same wavelength when the spectral radiance intensity of the actual object is equal to that of the blackbody. The brightness temperature is utilized to describe the radiation characteristics of the actual object itself. The equivalent brightness temperature spectrum $T(\nu)$ of the radiance spectrum is calculated based on Planck's radiation law:

$$T(\nu) = \frac{h\nu}{k \ln\{[L(\nu) + 2hc^2\nu^3]/L(\nu)\}} \quad (4)$$

where h is Planck's constant with a value of 6.625×10^{-34} J·S. c is the speed of light, with a value of 2.998×10^8 m/s. ν is the wavenumber with the unit of cm^{-1} . k is Boltzmann's constant, with a value of 1.380649×10^{-23} J/K. $L(\nu)$ is the radiance in terms of the wavenumber ν .

The diagram of bright temperature spectrum of Type 6 engine obtained by external field experiment is shown as follows. The wavebands of the characteristic peaks of the main components of the hot jet (carbon monoxide and carbon dioxide [26]) are displayed, and the positions of their infrared characteristic peaks are marked in the Figure 4.

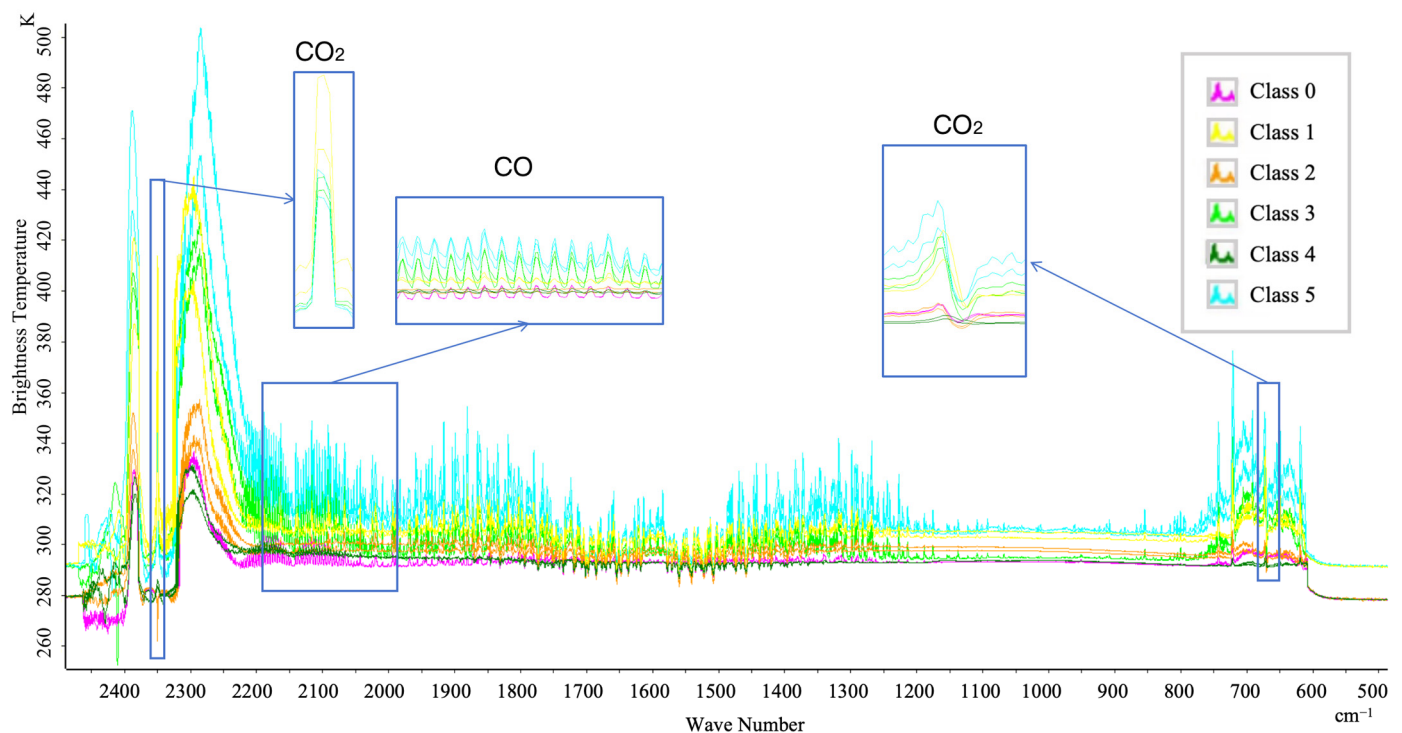


Figure 4. The infrared spectrum of the aeroengine hot jet (the characteristic positions of CO₂ and CO are marked in the figure).

A total of 1788 valid spectral data points were collected during the field experiment. In this study, adhering to standard deep learning dataset partitioning practices, we conducted random sampling at a ratio of 8:1:1 to generate the spectral training set, the validation set, and the test set. Spectral data within the mid-infrared range of 400–4000 cm^{−1} were systematically processed and resized to meet the network architecture requirements. Each spectrum comprises 7424 2D data points, serving as input for spectral feature extraction. Detailed parameters are provided in Table 4.

Table 4. Table of experimental aero-engines and environmental factors.

DATA SET	Data Volume	Category Proportion %						Select Range Data Volume
		C 0	C 1	C 2	C 3	C 4	C 5	
Training set	1432 (80%)	14.66	2.93	39.59	11.31	20.67	10.82	7424
Validation set	178 (10%)	10.67	1.69	43.26	10.11	26.97	7.30	7424
Prediction set	178 (10%)	15.17	1.69	38.2	10.67	20.22	14.04	7424

The original brightness temperature data were normalized to meet the needs of network model data processing, and the input data were normalized to the [−1,1] distribution by using the Min–Max Scaling variant. The normalized data $Data_{norm}$ and the anti-normalized data $Data_{original}$ (original brightness temperature data) represented by the meridional quantization are given by the following formula:

$$Data_{norm} = 2 \cdot \frac{Data_{original} - min}{max - min} - 1 \quad (5)$$

$$Data_{original} = \frac{Data_{norm} + 1}{2} \cdot max - min + min \quad (6)$$

In this context, $Data_{original}$ denotes the original spectral matrix, where each column represents a dimension (with the horizontal axis indicating wave number and the vertical axis representing brightness temperature). Vector min comprises the minimum values for each dimension, specifically $([min_{wn}, min_{bt}])$, while vector max consists of the corresponding maximum values for each dimension, namely $([max_{wn}, max_{bt}])$. Normalization of the dataset can be achieved through independent, element-wise operations on each dimension.

2.3. The Spectral Classification Network Structure Design Method

This section introduces the overall design of FTIR-SpectralGAN, the design of network composition, and the network training method.

2.3.1. Overall Network Design

The fundamental structure of the basic GAN comprises a generator (G) and a discriminator (D). The generator generates data conforming to the distribution of the training data by providing a set of random numbers (typically noise data following a Gaussian distribution) and eventually acquires the joint probability distribution of the data. The discriminator acquires the conditional probability distribution. GAN attains a Nash equilibrium through the interaction between the generator and the discriminator. Eventually, the generator acquires the data distribution of the original data, and the discriminator is capable of discriminating between authenticity. DC-GAN introduces CNN on the basis of GAN to enhance the generation quality and training stability. The previous research has demonstrated the superiority of CNN in spectral classification. Therefore, DC-GAN is selected as the fundamental model for spectral data augmentation to construct a network model applicable to the high-accuracy classification of aero-engine hot jet spectra. The paper aims to increase the value of the original data without actually augmenting the data in limited experimental tests, promoting a more stable and robust network and achieving a higher classification accuracy rate.

The FTIR-SpectralGAN designed is depicted as in Figure 5. Herein, the blue module represents the generator model, the green and purple modules stand for the discriminator models, and the yellow module indicates the data input. The input of the generator is random noise, which undergoes preliminary processing through a dense layer, undergoes shape adjustment and addition of Gaussian noise, and gradually generates the target data through multiple layers of transposed convolution, batch normalization, and LeakyReLU. At the final layer, the Tanh function is employed to output the generated pseudo-spectral data. The discriminator receives the normalized spectral data and the pseudo-spectral data generated by the generator and conducts feature extraction through multiple layers of one-dimensional convolution and LeakyReLU. During the convolution process, a Dropout layer is incorporated to prevent overfitting. The output section is divided into two branches. One branch utilizes the softmax function for classification prediction and the other branch adopts the sigmoid function for binary classification output of authenticity discrimination. Ultimately, the classification prediction results are utilized for performance evaluation.

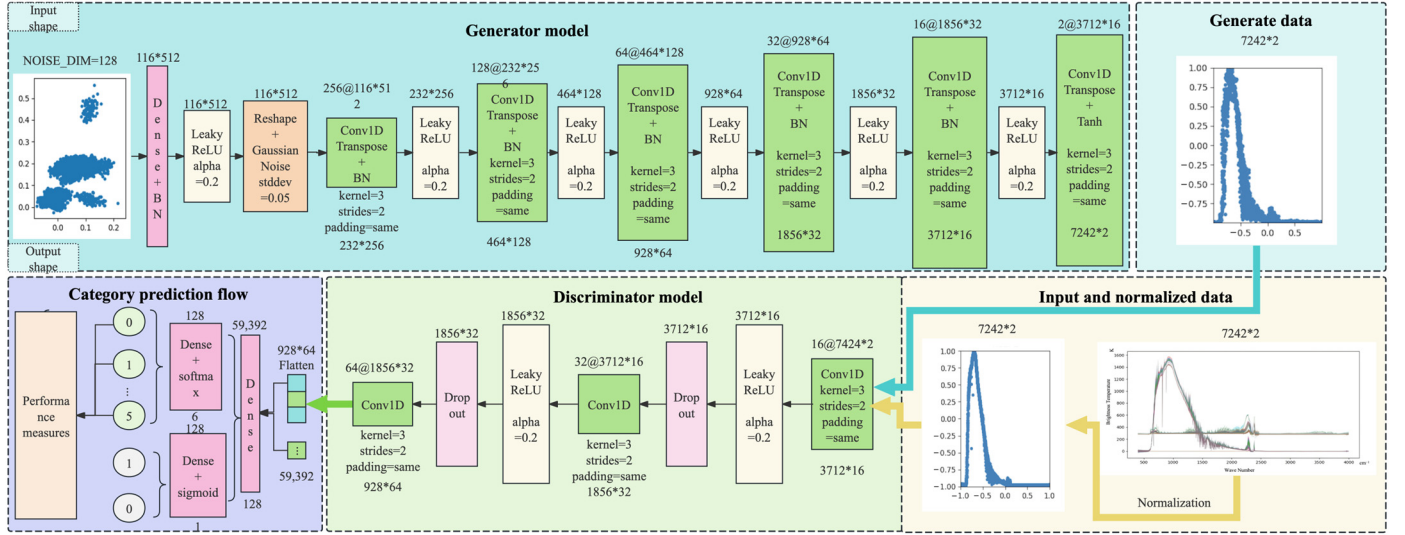


Figure 5. The overall design diagram of FTIR-SpectralGAN.

2.3.2. Network Composition Design

Based on the overall design in Figure 5, FTIR-SpectralGAN is composed of two crucial components: the generator and the discriminator. Herein, the generation network part generates spectral data by reconstructing random noise data, and the discrimination network predicts the category and authenticity based on the comparison between the original data and the generated data. The following elaborates on the generator and the discriminator separately.

1. Generative Network

The main components of the generative network are transposed convolutional layers, LeakyReLU layers, and batch normalization layers. Its composition schematic diagram is shown in Figure 6. The layer structure of the generator is given in the blue module in Figure 5.

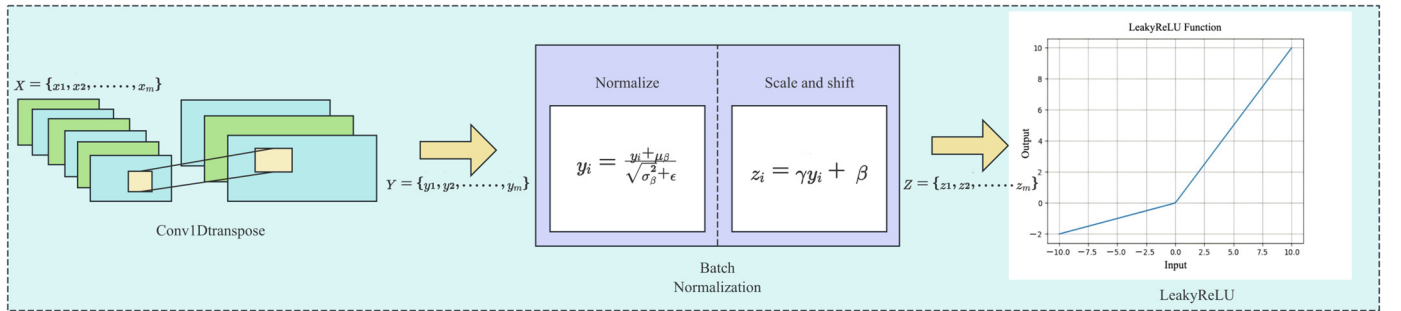


Figure 6. Diagram of the generative network structure module.

(1) 1D transposed convolutional layers (Conv1D transpose) [27]: The main operation of transposed convolution is to generate an output long sequence from the input sequence via interpolation and convolution operations, achieving the effect of upsampling and enhancing the resolution of the feature map. Suppose the input spectral sequence is $X = x_1, x_2, \dots, x_m$. Firstly, when the stride of the transposed convolution is $s = 2$, interpolation is conducted to obtain the sequence X' and its sequence length m' :

$$X' = \{x_1, 0, x_2, 0, \dots, x_m, 0\} \quad (7)$$

$$m' = (m - 1)s + 1 = 2m - 1 \quad (8)$$

It is filled in the same padding mode. The length n of the output sequence remains consistent with that of the input sequence m . Padding involves adding additional zeros on both sides of the interpolated sequence X' to ensure that the result of the transposed convolution meets the length requirement. When the convolution kernel size is selected as three in the model, p needs to be padded on both sides of the interpolated sequence, respectively, and the length of the padded spectral sequence is m'' .

$$p = \left\lceil \frac{k-s}{2} \right\rceil = 1 \quad (9)$$

$$X'' = \{0, x_1, 0, x_2, 0, \dots, x_m, 0, 0\} \quad (10)$$

$$m'' = m' + 2p = 2m + 1 \quad (11)$$

The filled sequence is subjected to a standard 1D convolution operation with the convolution kernel. The value at each output position y_i is expressed as

$$y_i = \sum_{j=1}^k X''_{i-j+1} \cdot w_j \quad (i = 1, 2, \dots, m) \quad (12)$$

In the formula, y_i represents the i th element of the output sequence, w_j indicates the weight of the convolution kernel, and k represents the size of the convolution kernel, which is set to three in the model. The length of the output sequence is also m and $Y = y_1, y_2, \dots, y_m$.

(2) Batch Normalization (BN) layer [28]: The BN layer can be represented as a learnable, parameterized network layer. The introduction of BN enables the network to learn to restore the feature distribution that the original network is supposed to learn. Suppose the output of the upper layer is $Y = y_1, y_2, \dots, y_m$, and the learning parameters are γ and β . The mean μ_β of the output of the upper layer is calculated as

$$\mu_\beta = \frac{1}{m} \sum_{i=1}^m (y_i) \quad (13)$$

where m represents the batch size of the training samples. In the model of the paper, m is set to 128.

Meanwhile, we calculate the standard deviation σ_β^2 of the upper-level data:

$$\sigma_\beta^2 = \frac{1}{m} \sum_{i=1}^m (y_i - \mu_\beta)^2 \quad (14)$$

The normalization processing of the output data Y can yield $\hat{y}_i$:

$$\hat{y}_i = \frac{y_i + \mu_\beta}{\sqrt{\sigma_\beta^2 + \epsilon}} \quad (15)$$

where ϵ is set to a small value close to zero to prevent the denominator from being zero.

The normalized data are reconstructed to obtain the output z_i :

$$z_i = \gamma \hat{y}_i + \beta \quad (16)$$

Herein, γ and β are learnable parameters. The BN layer acts before the nonlinear mapping. The BN layer is frequently employed to solve problems such as slow convergence and gradient explosion during network training. The incorporation of BN is also conducive to enhancing the accuracy of the model.

(3) LeakyReLU layer [29]: The LeakyReLU activation function introduces a leakage value in the negative half-interval based on the ReLU function, which can be expressed as

$$f(x) = \max(\alpha x, x) = \begin{cases} \alpha x, & \text{if } x < 0 \\ x, & \text{if } x \geq 0 \end{cases} \quad (17)$$

In the model, α is set to 0.2. The utilization of LeakyReLU can assist in resolving the dead ReLU phenomenon.

(4) Tanh activation function: In GANs, the generator employs Tanh as the activation function of the last layer. The output range of Tanh is $[-1, 1]$, which aligns with the standardized ranges of numerous data (such as $[-1, 1]$ or $[0, 1]$), facilitating the generator to output data that conform to the anticipated distribution. The Tanh function can be represented as

$$\text{Tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (18)$$

The gradient of the Tanh function is relatively large when approaching zero, which is conducive to rapidly adjusting the parameters of the generator in the initial stage of training and accelerating convergence. It is symmetrical with respect to the origin, which is beneficial for the generator to maintain symmetry when generating data and prevent the generated data from being biased towards one side.

2. Discriminator Network

The discriminator network needs to complete two parts: spectral classification and authenticity judgment. Its main component layers are convolutional layers, LeakyReLU activation layers, and Dropout layers. Finally, the features are connected by Flatten and Dense layers, and two Dense branches are used to perform, respectively, authenticity discrimination and spectral classification. The main components are shown by the green parts in Figure 7, and the classification part is completed by the purple background module in Figure 7. The structure and parameters are presented in Figure 5.

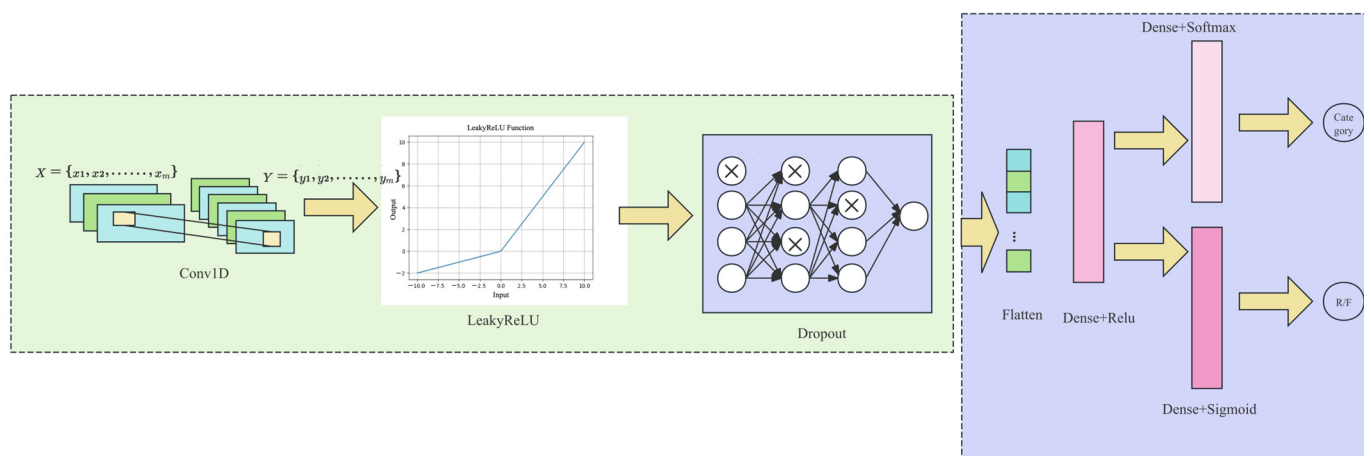


Figure 7. Diagram of the discrimination network structure module.

(1) Convolutional Layer (Cov1D) [27]: The Convolutional layer serves as a means for feature extraction in CNN, and it accomplishes linear and translation-invariant operations. Likewise, supposing the input spectral sequence is $X = x_1, x_2, \dots, x_m$, the convolution operation can be represented as

$$y_i = \sum_{j=0}^{k-1} w_j \cdot x_{i-s+j} + b \quad (19)$$

Herein, y_i represents the i th element of the output sequence of the convolutional layer, and k denotes the size of the convolution kernel, which is set as three in the paper. w_j refers to the j th weight of the convolution kernel, s represents the stride, which is also set as two, and b is the bias term. In the paper, the padding of the discriminator is set to be the same, indicating that the length of the output sequence remains unchanged and can also be expressed as $Y = y_1, y_2, \dots, y_m$.

Through convolution operations, we can extract the features of the data, enabling the enhancement of certain features of the original signal and the reduction in noise.

(2) Dropout Layer [30]: Dropout is a regularization method that randomly drops neurons with a certain probability and sets their output to zero, thereby reducing the risk of overfitting. Assuming the input sequence of the Dropout layer is $y = y_1, y_2, \dots, y_m$, Dropout can be expressed as

$$\tilde{y} = r \odot y, r \sim \text{Bernoulli}(1 - p) \quad (20)$$

where $\odot$ represents element-wise multiplication, r is the mask vector, with each element sampled from a Bernoulli distribution, and p is the dropout probability, which is set to 0.3 in the model of the paper, meaning that each upper neuron is set to zero with a 30% probability.

During the testing phase, to maintain the expected consistency, Dropout adjusts the output to

$$\tilde{y} = (1 - p) \cdot y \quad (21)$$

(3) Flatten Layer: The Flatten layer is used to flatten a multi-dimensional tensor into a one-dimensional vector, which can be expressed as

$$\text{Flatten}(Y) = [y_1, y_2, \dots, y_m] \rightarrow [y_{1,1}, y_{1,2}, \dots, y_{1,n}, y_{2,1}, y_{2,2}, \dots, y_{m,n}] \quad (22)$$

where $y_{i,j}$ represents the j th element of y_i .

(4) Dense Layer: The Dense layer is a fully connected layer. In the discriminator, the first Dense layer is combined with the ReLU [31] activation function. When the input is x , the weight matrix is W and the bias vector is b , and the output y can be represented as

$$y = \text{ReLU}(Wx + b) \quad (23)$$

where $\text{ReLU}(z) = \max(0, z)$.

In the layer of authenticity discrimination, the output y' is expressed via the sigmoid activation function:

$$y' = \sigma(z) = \frac{1}{1 + \exp(-z)} \quad (24)$$

In the classification layer, the output y' is expressed via the softmax activation function:

$$y''_i = \frac{\exp(z_i)}{\sum_j^n \exp(z_j)} \quad (25)$$

Here, n represents the number of categories, which is six in the paper. The output probability distribution satisfies $\sum_{i=1}^6 y''_i = 1$.

2.3.3. Network Training Methods

The training process of FTIR-SpectralGAN is accomplished by the alternating training of the discriminator and the generator [32], and an imbalanced training strategy is employed, in which the parameters of the generator are updated five times and the parameters of the discriminator are updated once to balance the performance of the generator and

the discriminator. The training and prediction processes of the network are presented in Figure 8, where the green background represents the training process and the blue background represents the prediction process. In the prediction process, the weights obtained after training are utilized for category prediction.

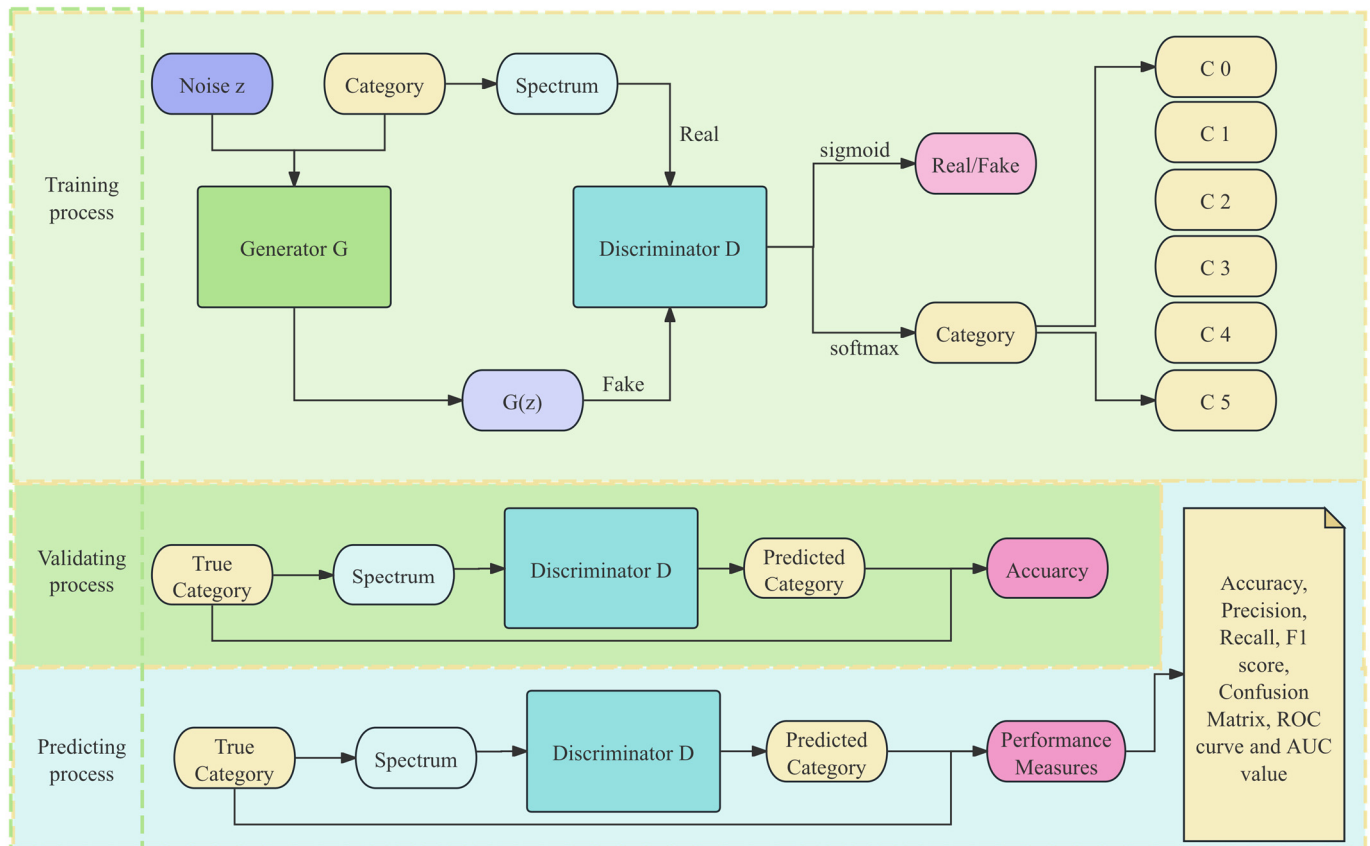


Figure 8. Graph of the training, validation, and prediction procedure of GAN.

(1) Weighted mixed loss strategy

The objective of the loss function of the generator is to maximize the predicted probability of the generated samples by the discriminator, namely to expect that the discriminator predicts the label of the generated samples as one (real). A binary cross-entropy function is employed, which can be represented as

$$Loss_G = E_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (26)$$

Here, $G(z)$ represents the samples generated by the generator, $D(G(z))$ is the discriminator's prediction for the generated samples, and $p_z(z)$ is the noise distribution input to the generator.

The objective of the discriminator is to maximize the discrimination accuracy for real samples while minimizing it for generated samples. The discriminator's loss function consists of two parts: authenticity loss and classification loss.

The authenticity loss part of the discriminator's output $Loss_{validity}$ is

$$Loss_{validity} = Loss_{real} + Loss_{fake} \quad (27)$$

$$Loss_{real} = E_{x \sim p_{data}(x)} [\log(0.9 - D(x))] \quad (28)$$

$$Loss_{fake} = E_{z \sim p_z(z)} [\log(0.1 + D(G(z)))] \quad (29)$$

where $Loss_{real}$ represents the loss of real samples and the output of the discriminator for this part should be close to the value of one. $Loss_{fake}$ represents the loss of generated samples, and the output of the discriminator for this part should be close to the value of zero.

For the loss of classification, the multi-class cross-entropy function is used for calculation, which can be expressed as

$$Loss_{class} = E_{x \sim p_{data}(x)} [\text{CrossEntropy}(y, C(x))] \quad (30)$$

Here, $C(x)$ represents the classification output of the discriminator and y stands for the true label.

The overall loss function of the discriminator can be expressed as

$$Loss_D = \alpha Loss_{validity} + \beta Loss_{class} \quad (31)$$

where α and β are weighting coefficients. Since the paper focuses more on the improvement of classification accuracy, α is set to one and β to four.

(2) Unbalanced training [33,34]

During GAN training, the training effects of the generator and the discriminator are different. Usually, the discriminator can achieve good results in a very short number of training iterations, which is very unfavorable for the training of the generator. In such cases, the gradient of the generator becomes very small (gradient vanishing), making it impossible for the generator to continue learning. Conversely, if the generator is too powerful, the discriminator cannot provide effective feedback, leading to training stagnation. Unbalanced training is a good solution, which adjusts the training frequencies of the generator and the discriminator to address the training disparity between them.

The core of unbalanced training is to adjust the update frequencies of the generator and the discriminator. Suppose the generator is updated k times and the discriminator is updated once. The training process of the discriminator can be described as

$$\theta_D \leftarrow \theta_D - \eta_D \nabla_{\theta_D} Loss_D \quad (32)$$

where η_D is the learning rate of the discriminator, which is set to LEARNING_RATE_D = 0.0005 in the model.

The training process of the generator can be described as when $i = 1, 2, \dots, k$,

$$\theta_G \leftarrow \theta_G - \eta_G \nabla_{\theta_G} Loss_G \quad (33)$$

where η_D is the learning rate of the discriminator, which is set to LEARNING_RATE_G = 0.0001 in the model. In each iteration, the discriminator is updated once while the generator is updated k times, with $k = 5$ in the model.

The essence of unbalanced training is to balance the gradient dynamics of the generator and the discriminator by adjusting their update frequencies. The unbalanced training strategy accelerates the learning of the generator. By allowing the generator to update multiple times, the quality of the generated samples can be improved more quickly. At the same time, it prevents the discriminator from becoming too strong. By reducing the update frequency of the discriminator, the generator has more opportunities to improve its generation ability.

(3) Label smoothing strategy

When discriminating between authenticity and falsehood, the smoothing strategy is to prevent overfitting and enhance the stability of confrontation. Suppose the label for true

and false classification is $y_{real} = 1$ and the label for the generated sample is $y_{fake} = 0$. After applying label smoothing, the target label is adjusted as

$$\begin{cases} y_{real}^{smooth} = 1 - \epsilon \\ y_{fake}^{smooth} = \epsilon \end{cases} \quad (34)$$

where ϵ is the smoothing coefficient, which is set to 0.1 in the paper, balancing the smoothing intensity and the clarity of the objective. And thus, the true sample label can be obtained as 0.9 and the generated sample label as 0.1.

(4) Optimizer

An optimizer is a method for finding the optimal solution of a model. The Adaptive Moment Estimation (Adam) [35] optimizer combines the advantages of momentum and RMSProp, using exponentially weighted averages to estimate the first and second moments. The first moment estimate (momentum) m_t of the gradient is calculated:

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t \quad (35)$$

where g_t is the gradient at the current time step and β_1 is the decay rate, typically set to 0.9.

The second moment estimate of the gradient (RMSProp) is calculated as follows:

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2 \quad (36)$$

where g_t is the gradient at the current time step and β_2 is the decay rate, usually set to 0.999.

To address this, deviation correction is carried out:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (37)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (38)$$

Finally, the updated parameters can be obtained as follows:

$$\theta_t = \theta_{t-1} - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t}} + \epsilon \quad (39)$$

where η represents the learning rate and ϵ is a constant, usually set to 10^{-6} . The Adam optimizer can effectively handle non-stationary objectives and sparse gradient issues by adaptively adjusting the learning rate for each parameter, and it is widely used in the training of deep learning models.

3. Experiment and Results

Section 3 comprises experimental tests and performance measures. It introduces the experimental environment and presents the experimental results and performance measures of the FTIR-SpectralGAN.

To study the effectiveness of FTIR-SpectralGAN on the measured aero-engine spectral dataset, in this paper, we conduct the FTIR-SpectralGAN spectral classification experiment. The experiment is carried out on a Windows 10 workstation (MSI Technology, Shanghai, China), which has 32 G of running memory, an Intel Core i7-8750H processor, and a GeForce RTX 2070 graphics card. The workstation is equipped with library environments such as python, tensorflow, and keras.

To evaluate the spectral experiment from multiple dimensions, the experimental performance measures consist of accuracy, precision, recall, F1 score, confusion matrix, receiver operating characteristic (ROC) curve, and AUC value. Among them, accuracy provides a

measure of the overall performance of classification, precision provides a measure of the accuracy of positive sample prediction, recall provides a measure of the ability to identify true positive samples, F1 score provides a measure of the balance between precision and recall, confusion matrix provides a detailed analysis of classification results, and ROC curve and AUC provide a measure of the overall performance of the model.

The specific parameters of the FTIR-SpectralGAN constructed in the experiment are given in Table 5.

Table 5. Model parameters for FTIR-SpectralGAN.

Layers	Parameter Settings
Input	input_shape = (7424, 2), num_classes = 6
Network parameter settings	EPOCHS = 500, BATCH_SIZE = 128, NOISE_DIM = 128, LEARNING_RATE_G = 0.0001, LEARNING_RATE_D = 0.0005, CHANNEL_1 = 16, CHANNEL_2 = 32, CHANNEL_3 = 64, CHANNEL_4 = 128, CHANNEL_5 = 256, CHANNEL_6 = 512
Generator	Dense((data_shape_x // 64) * CHANNEL_6), BatchNormalization(), LeakyReLU(), Reshape((data_shape_x // 64, CHANNEL_6)) CHANNEL_5 to CHANNEL_1: Conv1DTranspose(CHANNEL, kernel_size = 3, strides = 2, padding = 'same'), BatchNormalization(), LeakyReLU() Conv1DTranspose(2, kernel_size = 3, strides = 2, padding = 'same', activation = 'tanh')
Discriminator	CHANNEL_1 to CHANNEL_3: Conv1D(CHANNEL, kernel_size = 3, strides = 2, padding = 'same') LeakyReLU(), Dropout(0.3), Flatten()(x) Dense(128, activation = 'relu') Dense(num_classes, activation = 'softmax', name = 'class_output') Dense(1, activation = 'sigmoid', name = 'validity_output')

Based on the parameters in the table, we train the datasets and make label predictions, respectively, obtaining the experimental results as shown in Table 6, Figures 9 and 10.

Table 6. Experimental results of FTIR-SpectralGAN.

Evaluation Criterion Methods	Accuracy	Precision	Recall	F1-Score
FTIR-SpectralGAN	99.44%	99.76%	99.24%	99.49%

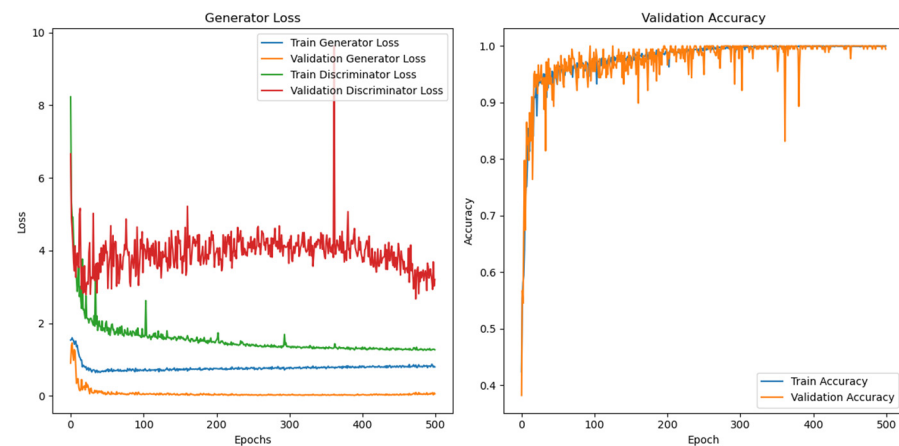


Figure 9. The loss function and accuracy change curves of the training set and validation set of FTIR-SpectralGAN. In the left figure, the blue line represents the loss of the generator in the training set, the orange line represents the loss of the generator in the validation set, the green line represents the loss of the discriminator in the training set, and the red line represents the loss of the discriminator in the validation set. In the right figure, the blue line represents the classification accuracy of the training set and the orange line represents the classification accuracy of the validation set.

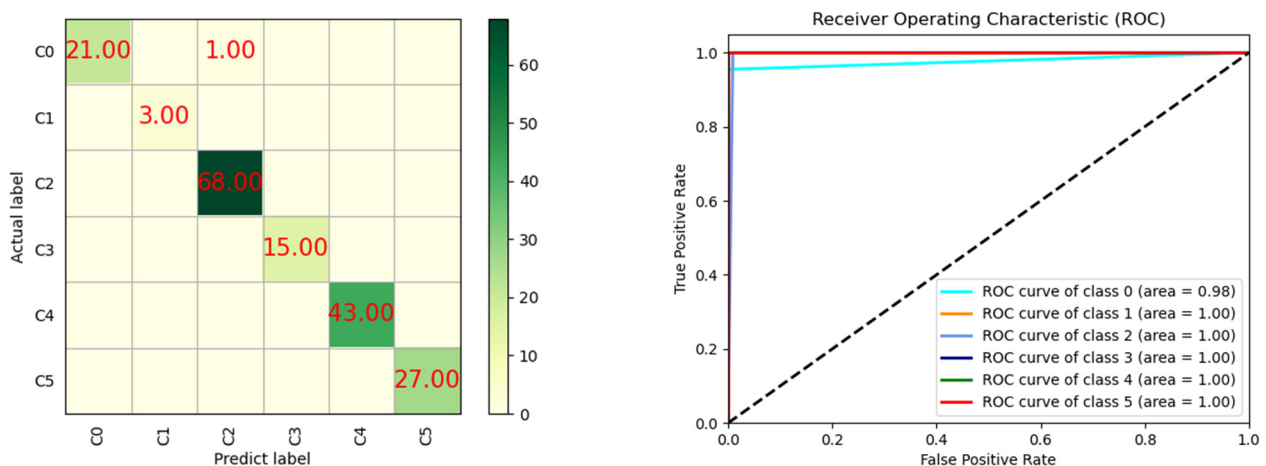


Figure 10. The confusion matrix and ROC curve graph of FTIR-SpectralGAN. In the right graph, the sky-blue line represents the ROC curve of C0, the orange line represents C1, the blue line represents C2, the dark blue represents C3, the green represents C4, the red represents C5, and the black dotted line represents the classification baseline.

Analysis of the experimental results of FTIR-SpectralGAN shows that the overall accuracy of the model on the six types of samples reaches a high accuracy rate of 99.44%, with most samples being correctly predicted; 99.76% precision indicates that the proportion of positive classes in the prediction of positive classes is very high, and the false positive rate is low; 99.24% recall rate indicates that the proportion of samples successfully identified as positive classes is very high, and the false negative rate is low; 99.49% F1 score represents that the model achieves a good balance between precision and recall. The confusion matrix shows that the model misclassifies C0 as C2. The FTIR-SpectralGAN model performs well in all performance measures, especially in accuracy, precision, recall rate, and F1 score, all of which are close to 100%. The confusion matrix shows that the classification effect on each category is excellent with almost no misclassification.

The loss curve can visually reflect whether the model is correctly optimizing the objective function. By analyzing the loss function change curve of FTIR-SpectralGAN, it can be seen that the losses of the generator and the discriminator are generally low and tend to be stable, indicating that the model achieves an optimal dynamic equilibrium. The

training loss of the generator drops rapidly and stabilizes, and the validation set shows a consistent trend but is slightly higher than the training loss, suggesting that the generator is continuously improving its output quality without obvious overfitting. The training loss of the discriminator initially drops rapidly and then stabilizes at a low value. A sharp peak appears in the validation set around 380 epochs, which might be due to the unstable quality of the generated data causing a temporary increase in the validation loss. However, this phenomenon occurred only once, and the loss recovered rapidly in the subsequent training, which belongs to the normal fluctuation of GAN training.

In the confusion matrix diagram, the horizontal axis represents the predicted labels and the vertical axis represents the true labels. The values on the diagonal indicate the number of samples classified correctly while the values off the diagonal represent the number of samples classified incorrectly. The analysis of the confusion matrix reveals that there are misclassification cases between C0 and C2, and one sample of C1 is predicted as C2 by the model. In addition, the remaining categories are all classified correctly. For a model with good classification performance, the ROC curve is closer to the top-left corner, and the AUC is used to measure the generalization ability of the algorithm. It is known that the AUC of category C0 is 0.98, indicating that its classification ability is relatively slightly weaker. The AUC of the remaining categories is all 1.00, suggesting that the classification performance of the model in these categories is extremely excellent.

The FTIR-SpectralGAN model performs well in all evaluation metrics. The confusion matrix shows that the classification effect of the model in each category is very good, with almost no misclassification cases. Since the accuracy rates of the training and validation data are almost the same and the loss of the model tends to be stable without excessive fluctuations, the problem of overfitting seems to have been effectively controlled. The model does not overfit on the training set but is able to generalize well to the validation set.

4. Discussion

4.1. Discussion on Comparative Experimental Results of Spectral Data Augmentation Methods

In this paper, we conduct experiments on the commonly utilized data augmentation methods to verify the superiority of the FTIR-SpectralGAN method proposed herein in enhancing the classification accuracy. We systematically compare several classical data augmentation techniques with CNN-based spectral classification methods, including rotation, scaling, translation, resampling, reflection, jitter, and dropout [36]. Simultaneously, the deep learning enhancement method, the convolutional variational autoencoder (Variational Autoencoder, CVAE [37]), is also employed for the experiments. In addition, we also conduct comparison with the spectral feature classification (PCA + SVM, CO₂ + XGBoost, CO₂ + Random Forest) methods.

We conduct a spectral classification comparison experiment on the spectral dataset using the parameters in Table 7. During the training of the network, the original data and the generated data were input into the network in a 1:1 ratio. The experimental results are shown in Table 8 and Figure 11.

Table 7. Parameter information of comparison algorithms.

Methods		Parameter Settings	Data Augmentation Methods	
			Methods	Parameter Settings
Data Augmentation	CNN	CHANNEL_1 to CHANNEL_3: Conv1D(CHANNEL, kernel_size = 3, strides = 2, padding = 'same') LeakyReLU(), Dropout(0.3) Flatten()(x) Dense(128, activation = 'relu') Dense(num_classes, activation = 'softmax') Dense(1, activation = 'sigmoid')	Rotation	Random rotation (0, 2 π)
			Scaling	Random scale (0.8, 1.2)
			Translation	Max translation = 0.1
			Resampling	Samples = 50
			Reflection	Random reflection
			Jitter	Noise level = 0.05 Decimal places = 2
			Dropout	Dropout rate = 0.1
Data Synthetic	CVAE	CHANNEL_1 = 32, CHANNEL_2 = 16, CHANNEL_3 = 8, CHANNEL_OUTPUT = 1 Encoded: (CHANNEL_1 to CHANNEL_3) Conv1D (kernel_size = 3, activation = 'Tanh', padding = 'same', kernel_regularizer = l2(0.01)), MaxPooling1D (2, padding = 'same') Latent space:Dense (z_mean),Dense (z_log_var), Lambda (z = z_mean + tf.exp (0.5 $\times$ z_log_var) $\times$ epsilon) Decoded: (CHANNEL_3 to CHANNEL_1) Conv1DTranspose (kernel_size = 3, strides = 1, activation = 'Tanh', padding = 'same'), UpSampling1D(2) Flatten = Flatten(),Dense (num_classes, activation = 'softmax') optimizer = tf.keras.optimizers.Adam (lr = 0.0001) loss = ['mse', 'sparse_categorical_crossentropy'], loss_weights = [0.5, 0.5] epochs = 500, batch size = 64		
Spectral Feature	PCA +SVM	PCA(n_components = 0.95), Svm = SVC(kernel = 'rbf', C = 10, gamma = 0.01)		
	CO ₂ +XGBoost	$a = [a_1, a_2]$ $a_1 = T_{v=2390\text{cm}^{-1}} - T_{v=2350\text{cm}^{-1}}$ $a_2 = T_{v=719\text{cm}^{-1}} - T_{v=667\text{cm}^{-1}}$	objective = 'multi:softmax' estimators = 500 estimators = 500	
	CO ₂ + Random Forest		estimators = 500	

Table 8. Classification experiment results of different comparison algorithms on spectral datasets.

Evaluation Criterion		Accuracy	Precision	Recall	F1-Score
Methods					
Data Augmentation	CNN	96.09%	98.72%	94.70%	96.18%
	Rotation + CNN	96.65%	98.89%	95.45%	96.79%
	Scaling + CNN	99.44%	99.80%	99.24%	99.51%
	Translation + CNN	96.09%	98.72%	94.70%	96.18%
	Resampling + CNN	96.09%	98.72%	94.70%	96.18%
	Reflection + CNN	96.09%	98.72%	94.70%	96.18%
	Jitter + CNN	96.65%	98.89%	95.45%	96.79%
	Dropout + CNN	96.09%	98.72%	94.70%	96.18%
Data Synthetic	FTIR-SpectralGAN	99.44%	99.76%	99.24%	99.49%
	CVAE	84.35%	84.85%	84.27%	83.85%
Spectral Feature	PCA + SVM	93.82%	90.51%	88.71%	89.34%
	CO ₂ + XGBoost	94.41%	91.75%	93.33%	92.43%
	CO ₂ + Random Forest	94.97%	92.38%	94.49%	93.2%

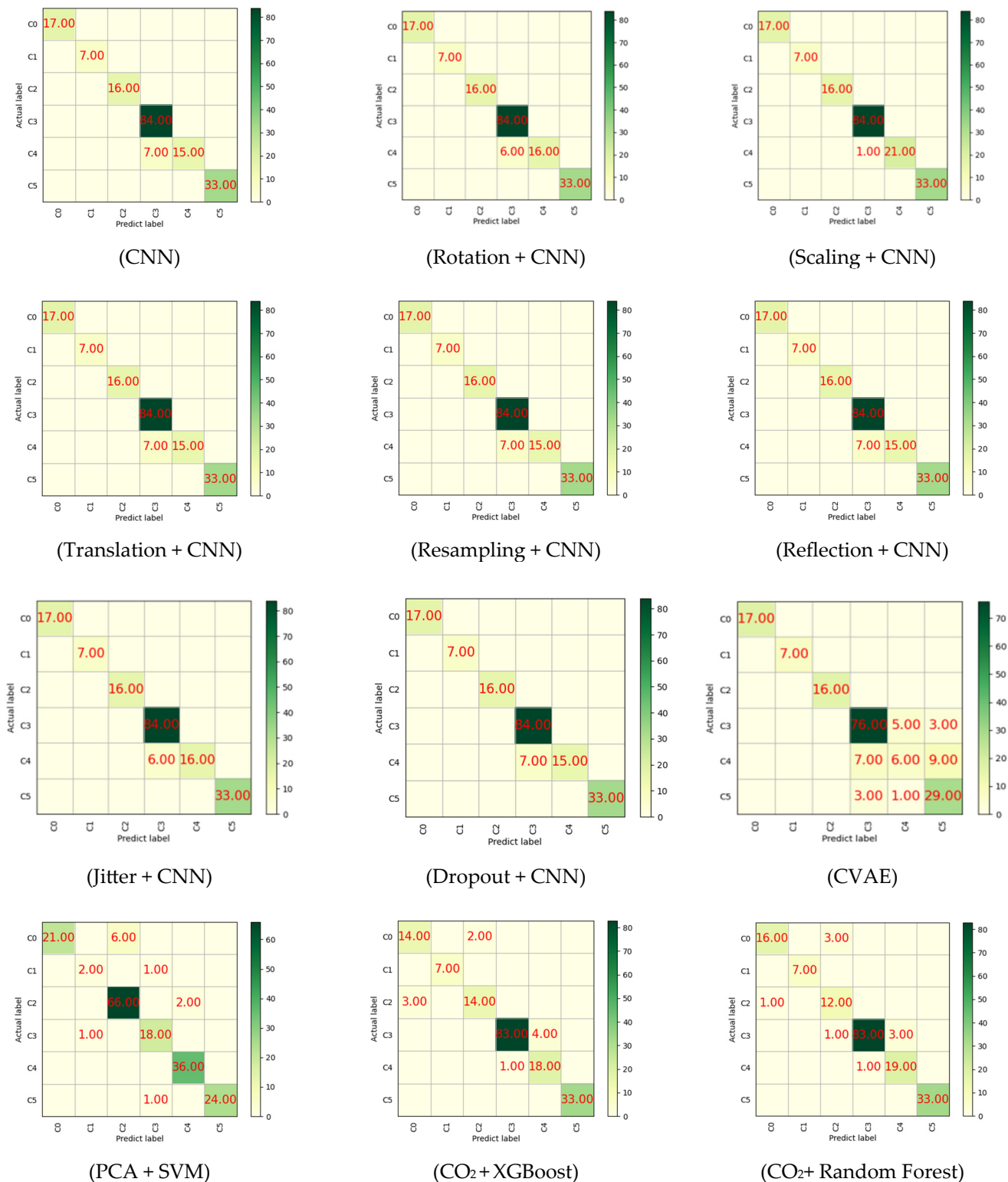


Figure 11. The confusion matrix diagram of the comparison algorithm where the abscissa is the predicted category and the ordinate is the true category.

According to the experimental results in Table 8, it is shown that among the data augmentation methods, Rotation, Scaling, and Jitter all contribute to improving the classification accuracy of CNN, reducing the initial CNN's misclassification. Among them, the scale change has the most significant effect on enhancing the model's classification ability. Other data augmentation methods do not improve the classification accuracy of CNN, nor do they interfere with the classification of CNN. The misclassifications are mainly concen-

trated between the fourth and fifth categories. In contrast, the classification results of the CVAE method are not satisfactory, with the quality of the generated data being relatively low, and it does not help improve the classification ability of the network. However, all three methods of spectral feature classification can achieve a classification accuracy rate of more than 93%, and there are relatively few misclassifications.

By analyzing the confusion matrices and ROC curves of each method, it can be observed that confusion occurs between C3 and C4 in CNN, with some C4 being predicted as C3 by the model. The Rotation, Jitter, and Scaling methods mitigate the confusion between C3 and C4 to some extent, among which Scaling contributes more significantly to correct classification. The other data augmentation methods for CNN do not exert a positive guiding effect on correct classification. The CVAE method exhibits more severe confusion among C3, C4, and C5, and the overall classification accuracy is not satisfactory.

The algorithm in the paper is trained with a ratio of 1:1 between the original data and the generated data, which influences the overall running time of the algorithm. The following table compares the training and prediction times of FTIR-SpectralGAN and each algorithm, as shown in Table 9.

Table 9. Test time comparison.

Methods	Train Time/s	Prediction Time/s
CNN	959.04	0.23
Rotation + CNN	2197.84	0.20
Scaling + CNN	2151.22	0.19
Translation + CNN	2181.78	0.18
Resampling + CNN	2089.91	0.19
Reflection + CNN	2111.77	0.20
Jitter + CNN	2159.05	0.19
Dropout + CNN	2123.13	0.19
FTIR-SpectralGAN	4661.46	0.35
CVAE	1790.76	2.25
PCA + SVM	1.1060	0.0588
CO ₂ + XGBoost	1.3269	0.5523
CO ₂ + Random Forest	0.8447	0.4293

In terms of training time, FTIR-SpectralGAN requires a longer training period due to its high model complexity, larger number of parameters, and generation of spectral data. Compared to the training time of FTIR-SpectralGAN, the CNN method has a shorter training time. After combining with classical data augmentation methods, the training time for CNN increases but remains relatively similar overall. The CVAE algorithm has a higher training speed. Regarding prediction time, most methods can complete predictions within one second. The prediction time of CVAE is slightly longer but still meets the requirements for real-time prediction. Under the characteristics of PCA and CO₂, the classical classifier methods have a very short training time. From the perspective of prediction time, the prediction time of most methods can meet the requirements of real-time prediction.

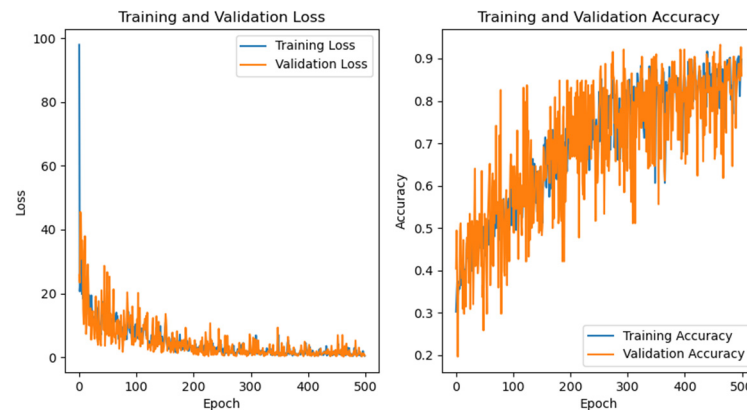
4.2. Discussion on the Ablation Experiment Results

(1) The effectiveness of synthetic data enhancement methods:

To verify the effectiveness of the spectral data augmentation method proposed in the paper, a CNN model with the same parameters as the discriminator is used to test the original spectral data. The experimental results are shown in Table 10 and Figure 12.

Table 10. Experimental results of spectral classification using the discriminator CNN.

Evaluation Criterion Methods	Accuracy	Precision Score	Recall	F1-Score
CNN	89.94%	97.06%	86.36%	86.85%

**Figure 12.** Loss and accuracy graphs of spectral classification using the discriminator CNN.

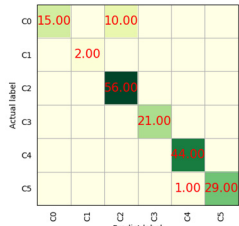
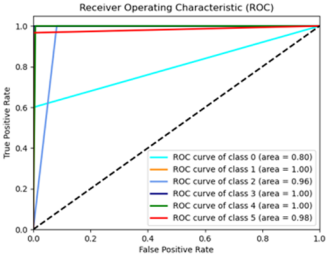
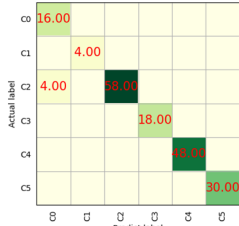
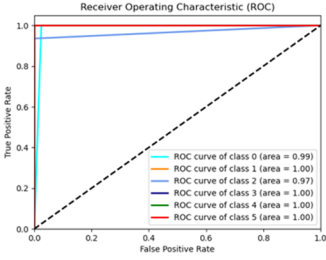
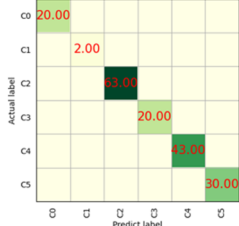
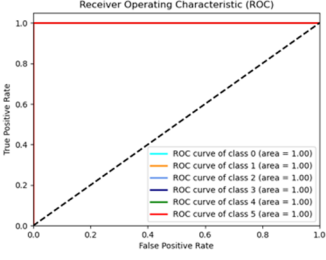
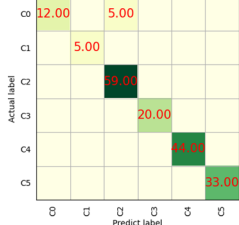
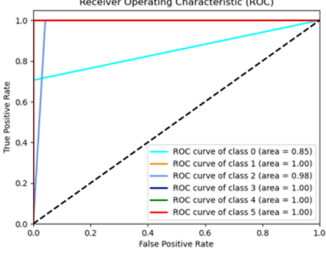
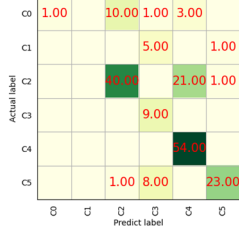
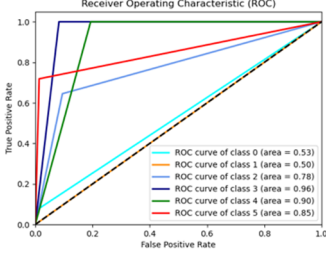
With regard to Table 10, this CNN model exhibits satisfactory performance on the majority of performance measures, particularly excelling in terms of accuracy. The confusion matrix reveals that the model has misclassification problems in certain categories, especially in C3 and C4. The F1 score indicates that the model strikes a favorable balance between precision and recall, and its overall performance is relatively stable. When comparing Table 10 with Table 6, it is evident that the FTIR-SpectralGAN approach yields higher classification data in all aspects than the mere utilization of the CNN, suggesting that the data augmentation method enhances the model.

Analysis of the loss and accuracy curves of this network shows that the curves present an oscillatory convergence, indicating that the model can learn the features of the data to a certain extent. However, the amplitude of the curve vibration is relatively large, and the model is not stable. The final classification accuracy after 500 training sessions approaches 90%. In contrast, in FTIR-SpectralGAN, the training accuracy and validation accuracy are basically consistent and gradually approach and stabilize, indicating that the model does not experience significant overfitting and has better generalization ability. Its performance on the validation set is reliable and has strong generalization ability. FTIR-SpectralGAN has a higher accuracy and converges faster. It is speculated that its discriminator effectively assists the generator in optimizing the data distribution, thereby directly improving the training and validation accuracy of the entire system.

(2) Optimizer selection effectiveness:

Commonly used optimizers include SGD, RMSProp, Adam, Adagrad, and Adadelta. Among them, RMSProp and Adam are more frequently employed. Different optimizers have varying adaptabilities to different tasks. The paper tests the backbone network with different optimizers on the dataset (setting the learning rates of the generator and discriminator as $\text{LEARNING_RATE_G} = 0.0001$ and $\text{LEARNING_RATE_D} = 0.0005$, respectively) for the FTIR-SpectralGAN spectral classification. The results are presented in Table 11 and Figure 13.

Table 11. Experimental results for different optimizers.

Optimizers	Accuracy	Training Time/s	Prediction Time/s	Confusion Matrix	ROC
SGD	94%	4434.86	0.74		
RMSProp	98%	4774.95	1.59		
Adam	100%	4515.55	0.70		
Adagrad	97%	4565.57	0.73		
Adadelta	71%	4599.13	0.64		

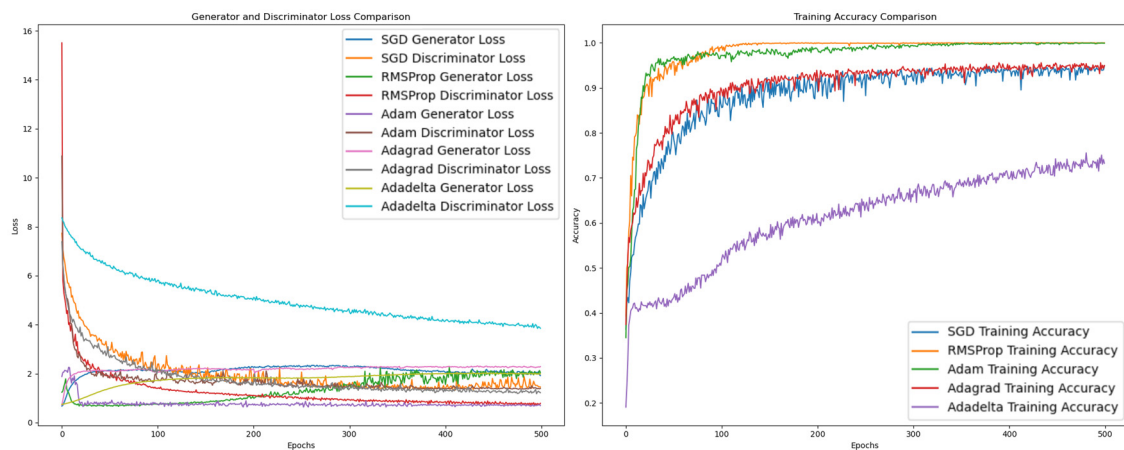


Figure 13. The curves of the training and validation loss functions and accuracy of the network on the dataset with different optimizers. In the left graph, the blue and orange curves represent, respectively, the generator and discriminator losses of the SGD optimizer; the green and red curves represent those of the RMSProp optimizer; the purple and brown curves represent those of the Adam optimizer; the pink and gray curves represent those of the Adagrad optimizer; and the light green and sky-blue curves represent those of the Adadelta optimizer. In the right graph, the blue curve represents the SGD optimizer; the orange curve represents the RMSProp optimizer; the green curve represents the Adam optimizer; the red curve represents the Adagrad optimizer; and the purple curve represents the Adadelta optimizer.

Analysis of the table data reveals that in terms of classification accuracy, among the five optimizers, SGD, RMSProp, Adam, and Adagrad all achieve a classification accuracy of over 90%, with Adam having the best accuracy, while Adadelta has a relatively lower classification accuracy. When comparing training time and prediction time, RMSProp has longer training and prediction times than the other optimizers. Overall, the Adam optimizer performs best in terms of both accuracy and time.

Analysis of Figure 12 reveals that the generator losses of various optimizers generally show a downward trend, indicating that the generator is gradually learning and improving the quality of the generated samples. In most cases, the discriminator losses rapidly decline initially and then stabilize, demonstrating that the discriminator is gradually developing better discrimination capabilities for the generator's output.

RMSProp and Adam optimizers perform best in terms of the convergence of generator and discriminator losses, while Adadelta and Adagrad exhibit significant oscillations in the early stages. RMSProp and Adam achieve the highest training accuracy and the fastest convergence speed, demonstrating higher learning efficiency. SGD and Adagrad lag slightly in accuracy improvement, with overall accuracy gradually approaching 95%. In contrast, the classification accuracy curve of Adadelta is not ideal. Although the accuracy curve gradually converges, the accuracy does not exceed 80%. Moreover, the loss value of the discriminator when Adadelta converges is relatively large. The possible reason is that the adaptive mechanism of Adadelta may fail to coordinate between G and D. Meanwhile, the Adadelta optimizer lacks an explicit momentum term. In the adversarial gradient scenario of GAN, it is prone to becoming stuck in local minima or oscillations, resulting in the difficulty of loss convergence.

By analyzing the confusion matrices and ROC curves of the five different optimizers, it can be seen that in this classification experiment, when using the SGD optimizer, 10 C0 samples were wrongly predicted as C2. In the RMSProp results, four C2 samples were wrongly predicted as C0. Adam did not produce any misclassification in this prediction. In the Adagrad results, five C0 samples were wrongly predicted as C2. Adadelta had

confusion in each category and had the worst performance. In conclusion, among the five optimizers, the Adam optimizer has the best performance.

Analysis of the experimental results can prove that the FTIR-SpectralGAN designed in this paper in the collaborative strategy of dynamic normalization strategies (Dropout, BN, Gaussian noise, label smoothing), adversarial training optimization (dual-task discriminator, unbalanced training), and synthetic data enhancement methods (generating synthetic data to expand and diversify the training data) reduces the overfitting problem that occurs in the small sample scene of spectra, realizes the systematic prevention and control of the overfitting risk, and achieves a higher classification accuracy rate.

5. Conclusions

This paper addresses the classification dilemma of limited spectral sample numbers of hot jet from aero-engines and designs the FTIR-SpectralGAN spectral classification algorithm. The SpectralGAN spectral classification network based on DCGAN proposed in this paper significantly improves the generation quality and feature discrimination ability of spectral data by adopting the method of enhancing classification with synthetic spectral data. Experiments show that this model achieves a high-precision classification performance of 99% on the FTIR spectral dataset. For the overfitting situation of the CNN classification network in the case of a limited number of samples, by means of dynamic regularization combinations (Dropout, BN, and adding Gaussian noise), stability-enhanced training strategies (dual-task discriminator, label smoothing, unbalanced training method, the generator produces high-fidelity synthetic spectra and in each round of training, the synthetic data: original data are trained at a ratio of 1:1) to increase data diversity and reduce the dependence of existing data on training data, a breakthrough in classification accuracy is achieved. Meanwhile, in this paper, we conduct comparative experiments on classic data augmentation methods (rotation, scaling, translation, resampling, mirroring, jittering, and dropping) and the deep learning enhancement method CVAE. The original data and the generated enhanced data are trained in a 1:1 ratio, and the classification accuracy and running time of each method are analyzed and compared, proving the effectiveness of the FTIR-SpectralGAN in classification. Ablation experiments are conducted by using only the CNN classification method of the discriminator and comparing experiments with different optimizers, proving the effectiveness of the FTIR-SpectralGAN method. Experiments prove that the algorithm framework proposed in present paper is effective and feasible and can achieve high classification accuracy, be more stable and robust, and complete the infrared spectral classification task of aero-engine hot jets.

The establishment of the aero-engine hot jet spectral dataset is in its infancy, and the insufficient data volume, different data resolutions, and data composition methods also pose new challenges for the research in present paper. The current research focuses on enhancing the spectral classification task through the utilization of synthetic data. However, it is important to note that the peak position, intensity, shape, and other characteristics of the FT-IR spectrum are intrinsically linked to molecular structure. Future research will concentrate on evaluating the reliability of synthetic pseudo-spectra and identifying instances where characteristic peak displacement exceeds reasonable limits. This can be achieved by incorporating domain knowledge constraints into the generator of GANs, such as introducing a spectral peak position regularization loss function or by employing conditional GANs to control the category specificity of generated spectra. Additionally, our current training strategy aims to improve classification accuracy but has not yet addressed the impact of unbalanced category distribution on results. We merely double the existing dataset to enhance classification performance. A more sophisticated training strategy should be designed to address the issue of unbalanced distribution. Finally, there is a risk that the

model may learn non-physically interpretable features. Future research should integrate the physical meaning of characteristic peaks to further enhance classification accuracy. The next step of the paper will focus on the spectral characteristics of different aero-engines for research in order to scientifically interpret the spectral characteristic positions of different substances in aero-engines and seek a more reliable spectral classification scheme.

Author Contributions: Formal analysis, Y.L.; investigation, S.D. and Z.L.; software, R.F. and F.L.; validation, Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China, grant number 62005320.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Li, S.; Song, W.; Fang, L.; Chen, Y.; Ghamisi, P.; Benediktsson, J.A. Deep Learning for Hyperspectral Image Classification: An Overview. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 6690–6709. [\[CrossRef\]](#)
2. Shi, W.; Koo, D.E.S.; Kitano, M.; Chiang, H.J.; Trinh, L.A.; Turcatel, G.; Steventon, B.; Arnesano, C.; Warburton, D.; Fraser, S.E.; et al. Pre-processing visualization of hyperspectral fluorescent data with Spectrally Encoded Enhanced Representations. *Nat. Commun.* **2020**, *11*, 726. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Nalepa, J.; Myller, M.; Kawulok, M. Hyperspectral Data Augmentation. *arXiv* **2019**, arXiv:1903.05580.
4. Wang, K.; Gou, C.; Duan, Y.; Lin, Y.; Zheng, X.; Wang, F.-Y. Generative adversarial networks: Introduction and outlook. *IEEE/CAA J. Autom. Sin.* **2017**, *4*, 588–598. [\[CrossRef\]](#)
5. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. GAN(Generative Adversarial Nets). *J. Jpn. Soc. Fuzzy Theory Intell. Inform.* **2017**, *29*, 177. [\[CrossRef\]](#)
6. Mirza, M.; Osindero, S. Conditional Generative Adversarial Nets. *arXiv* **2014**, arXiv:1411.1784.
7. Radford, A.; Metz, L.; Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv* **2015**, arXiv:1511.06434.
8. Miyato, T.; Kataoka, T.; Koyama, M.; Yoshida, Y. Spectral Normalization for Generative Adversarial Networks. In Proceedings of the International Conference on Learning Representations, International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
9. Karras, T.; Laine, S.; Aila, T. A Style-Based Generator Architecture for Generative Adversarial Networks. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019. [\[CrossRef\]](#)
10. He, Z.; Liu, H.; Wang, Y.; Hu, J. Generative Adversarial Networks-Based Semi-Supervised Learning for Hyperspectral Image Classification. *Remote Sens.* **2017**, *9*, 1042. [\[CrossRef\]](#)
11. Zhan, Y.; Hu, D.; Wang, Y.; Yu, X. Semisupervised Hyperspectral Image Classification Based on Generative Adversarial Networks. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 212–216. [\[CrossRef\]](#)
12. Zhu, L.; Chen, Y.; Ghamisi, P.; Benediktsson, J.A. Generative Adversarial Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 5046–5063. [\[CrossRef\]](#)
13. Ding, F.; Guo, B.; Jia, X.; Chi, H.; Xu, W. Improving GAN-based feature extraction for hyperspectral images classification. *J. Electron. Imaging* **2021**, *30*, 063011. [\[CrossRef\]](#)
14. Ranjan, P.; Girdhar, A.; Ankur, R.; Kumar, R. A novel spectral-spatial 3D auxiliary conditional GAN integrated convolutional LSTM for hyperspectral image classification. *Earth Sci. Inform.* **2024**, *17*, 5251–5271. [\[CrossRef\]](#)
15. Wang, J.; Gao, F.; Dong, J.; Du, Q. Adaptive DropBlock-Enhanced Generative Adversarial Networks for Hyperspectral Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 5040–5053. [\[CrossRef\]](#)
16. Li, W.; Chen, C.; Zhang, M.; Li, H.; Du, Q. Data Augmentation for Hyperspectral Image Classification With Deep CNN. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 593–597. [\[CrossRef\]](#)
17. Haut, J.M.; Paoletti, M.E.; Plaza, J.; Plaza, A.; Li, J. Hyperspectral Image Classification Using Random Occlusion Data Augmentation. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 1751–1755. [\[CrossRef\]](#)
18. Wang, W.; Liu, X.; Mou, X. Data Augmentation and Spectral Structure Features for Limited Samples Hyperspectral Classification. *Remote Sens.* **2021**, *13*, 547. [\[CrossRef\]](#)

19. Gao, H.; Zhang, J.; Cao, X.; Chen, Z.; Zhang, Y.; Li, C. Dynamic Data Augmentation Method for Hyperspectral Image Classification Based on Siamese Structure. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 8063–8076. [\[CrossRef\]](#)
20. Du, S.; Han, W.; Kang, Z.; Lu, X.; Liao, Y.; Li, Z. Continuous Wavelet Transform Peak-Seeking Attention Mechanism Conventional Neural Network: A Lightweight Feature Extraction Network with Attention Mechanism Based on the Continuous Wave Transform Peak-Seeking Method for Aero-Engine Hot Jet Fourier Transform Infrared Classification. *Remote Sens.* **2024**, *16*, 3097. [\[CrossRef\]](#)
21. Elaraby, S.; Sabry, Y.M.; Abuelenin, S.M. Super-resolution infrared spectroscopy for gas analysis using convolutional neural networks. In *Applications of Machine Learning 2020*; SPIE: Bellingham, WA, USA, 2020. [\[CrossRef\]](#)
22. Mao, M.; Cao, Y.; Ni, P.; Li, Z.; Zhang, X. Quantitative Analysis of Infrared Spectroscopy of Alkane Gas Based on Random Forest Algorithm. In Proceedings of the 2023 5th International Conference on Intelligent Control, Measurement and Signal Processing (ICMSP), Chengdu, China, 19–21 May 2023; pp. 1166–1170. [\[CrossRef\]](#)
23. Wang, Y.-H.; Liu, J.-G.; Xu, L.; Cheng, X.-X.; Deng, Y.-S.; Shen, X.-C.; Sun, Y.-F.; Xu, H.-Y. Qualitative Analysis of the Gas Detection Limit of Fourier Infrared Spectroscopy. *Acta Phys. Sin.* **2022**, *71*, 093201. [\[CrossRef\]](#)
24. Doubenskaia, M.; Pavlov, M.; Grigoriev, S.; Smurov, I. Definition of brightness temperature and restoration of true temperature in laser cladding using infrared camera. *Surf. Coat. Technol.* **2013**, *220*, 244–247. [\[CrossRef\]](#)
25. Homan, D.C.; Cohen, M.H.; Hovatta, T.; Kellermann, K.I.; Kovalev, Y.Y.; Lister, M.L.; Popkov, A.V.; Pushkarev, A.B.; Ros, E.; Savolainen, T. MOJAVE. XIX. Brightness Temperatures and Intrinsic Properties of Blazar Jets. *Astrophys. J.* **2021**, *923*, 67. [\[CrossRef\]](#)
26. Chu, P.M.; Guenther, F.R.; Rhoderick, G.C.; Lafferty, W.J. The NIST Quantitative Infrared Database. *J. Res. Natl. Inst. Stand. Technol.* **1999**, *104*, 59. [\[CrossRef\]](#)
27. Dumoulin, V.; Visin, F. A guide to convolution arithmetic for deep learning. *arXiv* **2016**, arXiv:1603.07285.
28. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv* **2015**, arXiv:1502.03167.
29. Xu, J.; Li, Z.; Du, B.; Zhang, M.; Liu, J. Reluplex made more practical: Leaky ReLU. In Proceedings of the 2020 IEEE Symposium on Computers and Communications (ISCC), Rennes, France, 8–10 July 2020. [\[CrossRef\]](#)
30. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
31. Glorot, X.; Bordes, A.; Bengio, Y. Deep Sparse Rectifier Neural Networks. *J. Mach. Learn. Res.* **2011**, *15*, 275.
32. Qin, Y.; Mitra, N.; Wonka, P. How does Lipschitz Regularization Influence GAN Training? In Proceedings of the Computer Vision—ECCV 2020, Glasgow, UK, 23–28 August 2020; Lecture Notes in Computer Science. Springer: Berlin/Heidelberg, Germany, 2020; pp. 310–326. [\[CrossRef\]](#)
33. Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X. Improved Techniques for Training GANs. *arXiv* **2016**, arXiv:1606.03498.
34. Brock, A.; Donahue, J.; Simonyan, K. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In Proceedings of the International Conference on Learning Representations, International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
35. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
36. Shorten, C.; Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 1–48. [\[CrossRef\]](#)
37. Chen, X.; Sun, Y.; Zhang, M.; Peng, D. Evolving Deep Convolutional Variational Autoencoders for Image Classification. *IEEE Trans. Evol. Comput.* **2021**, *25*, 815–829. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.